

Nama : Aulia Hannifah Chandra
Kelas : 2IA11
NPM : 50422299
Mata Kuliah : Pemrograman Berbasis Objek

Rangkuman Materi Pertemuan 1-6

A. Pengenalan Pemrograman Berbasis Objek

Pemrograman berorientasi objek (Object Oriented Programming atau disingkat OOP) adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Pemrograman berorientasi objek ditemukan pada Tahun 1960, dimana berawal dari suatu pembuatan program yang terstruktur (structured programming). Metode ini dikembangkan dari bahasa C dan Pascal. Pemrograman berorientasi objek bekerja dengan baik ketika dibarengi dengan Objek-Oriented Analysis And Design Process (OOAD).

a) Perbedaan Antara Pemrograman Berbasis Objek dengan Pemrograman Terstruktur

Pemrograman Berbasis Objek

- Menggabungkan fungsi dan data dalam kelas atau objek.
- Memiliki ciri Encapsulation (pengemasan), Inheritance (penurunan sifat) dan Polymorphism (perbedaan bentuk dan perilaku).
- Struktur program ringkas, cukup dengan membuat Objek dan class.
- Kode program sangat re-usable.
- Efektif digunakan untuk menyelesaikan masalah besar, karena OOP terdiri dari class-class yang memisahkan setiap kode program menjadi kelompok - kelompok kecil.
- Sulit diawal (karena harus membuat class).
- Eksekusi lebih cepat karena dieksekusi bersamaan.

Pemrograman Terstruktur

- Memecah program dalam fungsi dan data.
- Memiliki ciri Sequence (berurutan), Selection (pemilihan) dan Repetition (perulangan).
- Struktur program rumit karena berupa urutan proses dan fungsi-fungsi.
- Re-use kode program kurang.
- Tidak cocok untuk menyelesaikan masalah yang rumit, karena nantinya akan kesulitan menemukan solusi permasalahan ketika terjadi error.
- Mudah diawal, namun kompleks diproses selanjutnya.

- Eksekusi lebih lambat karena setiap perintah dikerjakan berurutan.

b) Class dalam PBO

Class adalah cetak biru atau blueprint dari object. Class digunakan hanya untuk membuat kerangka dasar. Sebagai analogi, class bisa diibaratkan dengan kendaraan. Kita tahu bahwa kendaraan memiliki ciri-ciri seperti merk, memiliki mesin, memiliki ban, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah kendaraan. Selain memiliki ciri-ciri, sebuah kendaraan juga bisa dikenakan tindakan, seperti: menghidupkan mesin atau mematikan mesin.

Contoh Program:

```
// Deklarasi Class
public class Kendaraan{
    // Ini Adalah Class, semua konstruktor, variabel, method berada disini
}
```

c) Objek dalam pemrograman

Objek adalah entitas yang memiliki atribut, karakter dan kadang kala disertai kondisi. Objek mempresentasikan sesuai kenyataan seperti siswa, mempresentasikan dalam bentuk konsep seperti merek dagang, juga bisa menyatakan visualisasi seperti bentuk huruf (font). Suatu objek didefinisikan berdasarkan namanya atau menggunakan kata benda, objek juga memiliki atribut dan metode. Pada dasarnya ada dua karakteristik yang utama pada sebuah objek , yaitu :

1. Setiap objek memiliki atribut sebagai status yang kemudian akan disebut sebagai state.
2. Setiap objek memiliki tingkah laku yang kemudian akan disebut sebagai behaviour.

Contoh: objek sepeda. Sepeda memiliki atribut (state): pedal, roda, jeruji, dan warna. Sepeda memiliki tingkah laku (behaviour): kecepatannya menaik, kecepatannya menurun dan perpindahan gigi sepeda.

d) Objek pada PBO

Untuk membuat objek, bisa menggunakan kata kunci new yang digunakan untuk membuat objek baru, selanjutnya menentukan 3 langkah untuk membuat sebuah objek. Yaitu: mendeklarasikan variable, membuat objek baru (Instansiasi) dan pemanggilan konstruktor.

Contoh Program:

```

public class Kendaraan{

    // Konstruktor Dengan Parameter
    public Kendaraan(String nama){
        System.out.println("Nama Kendaraannya Adalah "+ nama);
    }

    public static void main(String[]args){

        // Perintah untuk membuat objek jenis
        Kendaraan jenis = new Kendaraan("Pesawat Terbang");

    }

}

// Output = Nama Kendaraannya Adalah Pesawat Terbang

```

B. Inheritance

Pewarisan merupakan sebuah bentuk “penggunaan kembali” (reusability); dimana class baru dibuat dari class yang pernah ada yang (biasanya) ditambah fasilitasnya. Pewarisan tunggal (single inheritance) merupakan pewarisan dari satu class pokok (induk). Pewarisan ganda (multiple inheritance) merupakan pewarisan dari dua atau lebih class pokok.

Contoh untuk memanggil constructor milik superclass-nya :

```

super ()

super (parameter)

```

Contoh untuk memanggil atribut dan method milik superclass-nya :

```

super.namaAtribut

super.namaMethod(parameter)

```

C. Encapsulation

Enkapsulasi (encapsulation) merupakan cara untuk melindungi property (atribut) / method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program. Cara untuk melindungi data yaitu dengan menggunakan access modifiers (hak akses). Ada 4 hak akses yang tersedia, yaitu default, public, protected, private

No	Modifier	Pada class dan interface	Pada method dan variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method-method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method-method yang sepaket
4	private	Tidak bisa diterapkan	Tidak dapat diakses dimanapun kecuali oleh method-method yang ada dalam kelas itu sendiri

Aksesabilitas	public	private	protected	default
Dari kelas yang sama	Ya	Ya	Ya	Ya
Dari sembarang kelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari sembarang kelas di luar paket	Ya	Tidak	Tidak	Tidak
Dari subkelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari subkelas di luar paket	Ya	Tidak	Ya	Tidak

D. Polymorphism

Overloading adalah diperbolehkannya dalam sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda.

```

1 public class Overloading {
2     public void Tampil(){
3         System.out.println("I love Java");
4     }
5     public void Tampil(int i){
6         System.out.println("Method dengan 1 parameter = "+i);
7     }
8     public void Tampil(int i, int j){
9         System.out.println("Method dengan 2 parameter = "+i+" & "+j);
10    }
11    public void Tampil(String str){
12        System.out.println(str);
13    }
14
15    public static void main(String a[]){
16        Overloading objek = new Overloading();
17        objek.Tampil();
18        objek.Tampil(8);
19        objek.Tampil(6,7);
20        objek.Tampil("Hello world");
21    }
22 }

```

Overriding method adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk (mendefinisikan kembali) method superclass-nya. Contoh overriding method dapat dilihat pada subclass “Mobil” yang mendefinisikan kembali method keterangan() dan hapus() dari class “Kendaraan”.